

Review of Modern Virtual Reality HMD Devices and Development Tools

Aleksandar Jovanović¹ and Aleksandar Milosavljević²

Abstract - The paper describes the development of virtual reality with an emphasis on HMD devices and tools for application development. Firstly, we will describe popular VR HMD devices for PC, mobiles, and consoles. Secondly, we will describe the development environment for native, multi-platform and web applications. Next, through the development of VR software application, we will introduce best practices for VR development and user experience standards.

Keywords - Virtual reality, HMD, Software development, Game engine.

I. INTRODUCTION

In the mind of most people idea of virtual reality is connected with imaginary worlds from sci-fi movies or novels. By definition from 1993 virtual reality is: "Virtual reality refers to immersive, interactive, multi-sensory, viewer-centered, three-dimensional computer-generated environments and the combination of technologies required to build these environments" [1]. The first example of virtual reality appeared in 1938 with a device for watching stereoscopic image. However, the first stereoscopic device which brought attention to the public was a View-Master from 1939. Today, the stereoscopic device represents a fundamental pattern for making low-budget VR HMD devices. The first concept for a modern HMD (head-mounted display) device that has attracted attention was the Ultimate display from 1965. The concept is made by Ivan Sutherland, whose idea was to develop a device that will use a computer-generated graphic for creating interactive worlds. During that period the term VR was not yet known, computers were in the early stage of development and the forerunners of VR devices operated using cameras [2]. The first VR device which used computer hardware and object creation in real time appears in 1968 called Sword of Damocles. Creators of this device were Ivan Sutherland and his student Bob Sproull. During the 90's there were unsuccessful attempts of VR devices like an iGlasses, Nintendo Virtual Boy, Cybermaxx, and VFX-1 [3].

The first modern VR device was created in 2012 from young engineer and first CEO of company Oculus VR, Palmer Luckey. In August 2012, Oculus VR launches Kickstarter campaign about cutting edge technology called Oculus Rift. A social network company Facebook acquired Oculus VR for 2 billion dollars with a vision that VR will be the next big thing after mobile phones [4].

¹Aleksandar Jovanović is with the Faculty of Electronic Engineering at University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: aleksandarjovanovic@elfak.rs

²Aleksandar Milosavljević is with the Faculty of Electronic Engineering at University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: aleksandar.milosavljevic@elfak.ni.ac.rs

The goal of the paper is to showcase the potential of VR technologies. The paper will provide information regarding modern VR devices and software development techniques and tools. After, it will introduce an important concept for VR software development which includes best practices and standard for high-quality VR experience. High-quality VR experience is essential for building next generation product. Also, there is a demand on the market for VR developers and experts in HMD technologies like Oculus Rift and Google Cardboard. Regarding the concept proposed in this paper current and future developers will have a brief overview of the current state of technologies and knowledge to overcome beginning steps in VR developer carriers. The paper is organized in the following way. Firstly, we will describe modern VR HMD devices like Oculus Rift, Gear VR, Google Cardboard and HTC Vive. Secondly, we will describe VR application development for PC, mobile, and web-based solutions. Also, compare HMD devices for the VR application development. Thirdly, we will describe VR application *Ghost Hunt*. At the end, we will introduce best practices and user experience standard for VR development.

II. VR HMD DEVICES

The presence of the modern generation of virtual reality starts from 2012 and until now VR dedicated companies and industry leaders developed their own HMD devices and software development environments. HMD is a type of computer display device that is worn on the head or is built in as part of a helmet. The most popular VR HMD devices for smartphones, PC, and game consoles are Google Cardboard, Oculus Rift, Playstation VR, HTC Vive, Samsung Gear VR, Avegant Glyph, and Razer OSVR.

There are three types of VR devices [5]. The first type is Mobile VR which includes a portable device for a smartphone like Google Cardboard and Samsung Gear VR. This is the more intuitive approach for users because there is no need for PC device, additional settings, and cables for activating the application on the smartphone. The second type is a standalone device for PC desktop devices like Oculus Rift and HTC Vive. Unlike a first type, they are more optimized and provide a better experience from performance and graphic rendering perspective. The third type is VR for game consoles like Playstation VR.

The basis for all of modern VR devices are lenses, a display for watching stereoscopic pictures and inertial measurement unit (IMU). The lens is coming with different characteristic, for example, Oculus Rift has two lenses A and B. The difference is the distance between the lens and display device. The information is displayed on the screen using the graphic processing unit (GPU). The GPU allows devices to operate

with information and images which require high graphics performance units. Stereoscopic or 3D image creates a sense of depth by sending spatial information that creates a sense of reality. IMU is an electronic device which combined a sense of motion, such as an accelerometer, a gyroscope, a compass, for determining the position of the HMD. IMU has a very important role in displaying virtual environment because it has to calculate a rapid movement of users and provide a realistic experience [6]. Illustration of HMD device is shown in Fig. 1.

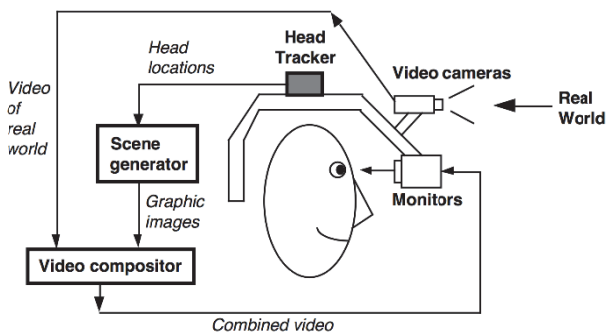


Fig. 1. Illustration of HMD device [7]

Below will be described VR devices for smartphones, PCs and game consoles [8]. Today, smartphones contain all sensors (e.g. IMU) necessary for the realization of the HMD except lenses for creating stereoscopic images. Google Cardboard is a box with lenses and without any electronic components, designed by Google [9]. Users can create Cardboard using DIY (Do It Yourself) specification. Cardboard devices allow the 90-degree field of view without any effect of distortion of the barrel and update of 60 frames per second (FPS) which is much less in comparison with standalone devices. It has support for iOS and Android smartphones. During January 2016 has published more than 1,000 applications and delivered more than 5 million units for Google Cardboard. Also, Google has created the next generation mobile platform Daydream that represents a high-quality VR for daydream-ready devices.

Oculus Rift is the first HMD in a new era of virtual reality. Like most VR devices requires use in seated or standing position. Oculus Rift has a display resolution of 2160x1200, refresh rate of 90fps, 100-degree field of view, motion tracking with a webcam, mounted headphones, Xbox integration, and Oculus touch. Also, Oculus has very good support for application development and Oculus store. The first commercial version (CV1) was released in March 2016 and sold during the pre-order phase [10].

Gear VR is a device created by Samsung and Oculus. Gear VR has support for barrel distortion effect and the latest Oculus technology for motion tracking. Samsung has released Gear VR *Innovator Edition* at the end of 2014, which supported only Samsung Note 4. Later versions add the support for Samsung Galaxy smartphone series after S6. Gear VR has an AMOLED screen resolution of 2560x1440 (1280x1440 per eye). Gear VR has its own IMU, which has less delay than the version on smartphones. Have external controls for interacting with the inner smartphone, because the

smartphone is inside the case and it is impossible to have a direct contact [11].

The next device is HTC Vive created by HTC and Valve companies. Based on the current specification is leading systems on the market. HTC Vive, unlike Oculus Rift, allows user's movement in space. Users who cannot move in a physical space can use the teleport technique which is very common in VR applications [12]. Teleport can help all users to freely explore the world around them. In addition, it should be noted that the teleport can have a negative impact on the user, so it's necessary to implement fade in/out effect when changing locations so users would not have felt a sudden transition to a different location [13]. HTC Vive supports a screen resolution of 2160x1200 and uses two sensors that are placed on the wall in order to monitor user movement [14].

III. DEVELOPMENT ENVIRONMENT

Before the beginning of the development, users must choose development environment. They can be divided into the following groups: native development, game engine tools and web-based solutions [5].

Native development includes drivers and the software libraries such as the Win32 library for C++ applications and Java libraries that are used in conjunction with the operating system. VR companies develop an SDK (software development kit) for most popular platforms and interface for accessing device hardware components. Native development of applications is the more flexible and optimized, but it requires a lot of time, especially if you want to develop multi-platform applications. Because of this, most developers use multi-platform development tools like game engines.

The game engine is an integrated development environment (IDE), primarily for the development of video games. It is used as middleware for accessing low-level elements of HMD devices. Also, game engines allow creating applications using higher programming languages. Due to its features game engines have become the basis not only for video games but also for other types of 3D applications. The game engine tool provides export to multiple platforms, visualizer, support built-in 2D and 3D graphics, physics engine, sound processing, support for scripting, animation, artificial intelligence (AI), networking, etc. The most famous commercial game engine tools are Unreal Engine 4 (a programming language for development is C++), CryEngine 3 (Lua), Unity 4 and 5 (C# Mono and JavaScript), Amazon Lumberyard (C++ or Lua), and others [15]. Regarding open-source solution most popular are libGDX (Java), Xenko (C#), Torque 3D (C++, TorqueScript), Urho3D (C++), OGRE (C++, Python, Java, C#), etc.

Web browsers include the development of multi-platform applications using HTML5, WebGL, and JavaScript technologies. An example is the development of applications using JavaScript 3D libraries like Three.js or Babylon.js that encapsulates WebGL library. Google Chrome and Mozilla Firefox Nightly browser have versions that support VR applications. Mozilla started the development of VR plugins for Firefox browser. Then Google and Mozilla have joined forces to develop a new web standard for VR called WebVR.

WebVR JavaScript API provides access to the user's web browser via the Oculus Rift and Google Cardboard devices. Regarding WebVR, Mozilla supported domain-specific language A-Frame for WebVR based applications[16].

Regarding the development of VR applications most popular tools are Unity and Unreal game engines. Unity is a tool for developers and designers who want to develop games fast and simple. The main focus is on the game business logic while the rest is managed by the game engine. Unity appeared in 2005 for the Mac OS and later was added support for Window and Linux systems. The core engine is written in a C++ development environment and graphical user interfaces in a C# programming language. For scripting, Unity uses MonoDevelop IDE that comes with the Unity game engine. In addition to video games, Unity is used to develop learning simulations [17]. From a point of software architecture, Unity can affect performance, unlike native solutions. Also, the Unity asset store provides a free content that includes graphic models, environment and programming scripts. Unity game engine provides support for all major VR platforms.

Regarding mentioned information from the paper, technology comparison is given in Table 1 for Oculus Rift, Gear VR, and Google Cardboard. Based on information from Table 1 Oculus Rift provide a high-quality experience for desktop PC and Unity integration. Below will be introduced Oculus Rift VR tech demo application.

TABLE I
TECHNOLOGY COMPARISON

Feature/device	Oculus	Gear VR	Cardboard
Native support	Oculus SDK, C/C++ API	Oculus Mobile, C++/NDK	Cardboard SDK, Java and OpenGL ES
Unity	Unity packet for desktop	Unity packet for Android	Unity SDK for Cardboard
Web programming WebGL	Unity WebGL or WebVR API	Unity WebGL or WebVR API	Unity WebGL or WebVR API
Platform	Desktop (Windows)	Android	Android and iOS
Experience	High-quality desktop VR	High-quality mobile VR	Low-quality mobile VR, depend on mobile devices
Company support	Oculus VR and Facebook	Samsung and Facebook	Google

IV. DEVELOPMENT OF VR APPLICATION

VR Oculus Rift tech demo application *Ghost Hunt* is a first-person survival horror video game[18]. Horror games are very popular on VR platforms, especially on desktop based

VR. The game was developed using the Unity game engine with support for mouse, keyboard or controller. The atmosphere of the game is shown in Fig. 2. *Ghost Hunt* is developed for the purpose of this paper and research of VR application development. Regarding operating system and hardware configuration, *Ghost Hunt* was developed on Windows 8.1 Pro, Asus N53S Notebook with Intel Core i3-2350M 2.30 GHz processor, 4GB RAM, and Nvidia GeForce 610 2GB graphic card. Experience with VR devices and software tools is not so much different if we compare with the standard game development. However, VR solutions require knowledge regarding product quality and user experience development.



Fig. 2. Sample screenshot from the game *Ghost Hunt*

Regarding development of VR high-quality solutions below will be introduced best practices and standards for development based on practical experience from the video game *Ghost Hunt*. The VR user interface is significantly different from standard 3D applications. It is important to know the resolution of the device, there are standards for Oculus Rift CV1 2160x1200 (1080 x 1200 per eye) desktop device and Gear VR 2560 x 1440 (1280 x 1440 per eye) mobile devices. Based on this we can create a user experience and make the environment more intuitive for consumers. Regarding optimization techniques, it is recommended to use Anti-Aliasing (AA) for a desktop device. AA technique is not recommended for most mobile devices. Regarding object rendering of VR scene, the consumer may notice visible pixels. In this case, the objects are scaled using technique RenderScale. RenderScale technique reduces performance but increases image quality. It is recommended to use low-poly mesh models for the rendering of the scene, especially with applications for the mobile platform. A game engine such as Unity has the option Level of Detail (LOD) which enables optimization of polygons in relation to the distance. Lightmapping is a tool that can significantly increase product performance. Lightmapping disables dynamic lighting and instead uses static lighting, which renders the scene at the beginning of creation. Throughout the life cycle of the scene, there will not be any additional changes. Occlusion Culling is a technique used in video games that use large scenes with a large number of objects, the idea is to use this technique for rendering objects that are currently visible to the user. All listed techniques are fully adaptable for VR experience on all major platforms.

Regarding UX/UI standards, there are four types of user interface for VR[19]: Non-diegetic UI describes elements that

do not belong to the virtual world, e.g. HUD (head-up display). This type of UI is poorly used in VR applications, primarily because of user focus. Spatial UI describes an element that is part of the VR world. In relation to non-diegetic, they have a position that is not accompanied by the user. Diegetic UI describes an element that follows some object in the scene. A good example is the energy of the enemy to the scene. Meta UI describes the effect on the scene that is not part of the VR world. For example, change of color if the user loses energy. Diegetic UI elements were used in video game *Ghost Hunt*. The main character uses diegetic element in a form of a smartphone. Also, smartphone flash effect represents the meta UI element on the player screen.

For adapting a standard 3D video game into VR there are a few simple steps from the Unity game engine. For Oculus Rift, the user needs a special camera rig that moves directly in Unity from Oculus integration package. Also, the user needs to enable *Virtual Reality Supported* check box in Player Settings. A similar way is for Cardboard application using the Unity SDK for Cardboard. For HTC Vive, the user needs to get a Steam account, install the runtime on the development machine, then go into Unity and download the SteamVR camera rig from the asset store and attach it to existing camera. At the end disable all in-game camera in favor of the new VR camera. All listed packages and features are free for download and development.

Because VR is in the early stage of development there is the possibility of error on major operating systems like Windows or Oculus SDK legacy runtime. It is recommended to use the latest version of Oculus SDK. The latest version of Oculus SDK supports game engine integrations for Unity, Unreal, WebVR, native and mobile applications. Before a developer publishes an application, the solution should follow best practice for best VR experience defined by Oculus team. After submission, the application will be reviewed by Oculus team. For example, if the application has the option to move the position of the head by y-axis using a mouse or controller, it will be returned for correction.

Regarding the best practices for mobile VR, developers should consider mobile application Cardboard Design Lab. The mobile platform is limited regarding user interaction and a novel approach to increase user experience and interaction is to use haptic devices.

V. CONCLUSION

This paper discussed the development of VR technology, with a focus on use cases and implementations. Firstly, VR HMD devices were described. Secondly, we describe the development environment for VR, Unity game engine and compare HMD devices. Thirdly, the development of the VR application was described using best practices and UI/UX standards. VR devices and software tools are ready for serious software development. However, software developers should be familiar with best practices if they want to achieve high-quality VR experience.

In the future, we can expect that VR moves forward, primarily because of the great interest of the companies and the community of users. Virtual and augmented reality

devices like Oculus Rift, Leap Motion or HoloLens have been increasing popularity among developers. The new interdisciplinary field will be created through the connection of VR with education, medicine, art, automotive, and architecture. Game engines are becoming increasingly popular, especially because it does not require financial investment and provides a simple interface for operation.

Regarding future research, there is a potential for the use of VR in education, communication and simulation development. VR platform can help in the development of inclusion, a community of users who will be ready to cooperate regardless of their limitations. Future research will focus on use cases that can improve people skills using modern VR technologies.

REFERENCES

- [1] C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE", *20th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH'93*, pp. 135-142, 1993.
- [2] T. Mazuryk, and M. Gervautz, "Virtual Reality History, Applications, Technology and Future", *Technical Report TR-186-2-96-06*, 1996.
- [3] P. Bouvier, F. De Sorbier, P. Chauderac, and V. Biri, "Cross-Benefits Between Virtual Reality and Games", *Computer Games Multimedia and Allied Technology Conference*, 2008.
- [4] E. Guirado Carmen, and G. Castro Carmen, "Crowdfunding as an Open Innovation for Co-Creation", *Strategic Approaches to Successful Crowdfunding*, 2016.
- [5] T. Parisi, "Learning Virtual Reality", *O'Reilly Media*, 2015
- [6] M. Tawadrous, D. Rojas, B. Kapralos, A. Hogue, and A. Dubrowski, "The Effects of Stereoscopic 3D on Knowledge Retention Within a Serious Gaming Environment", *Multimedia Tools and Applications*, March 2016.
- [7] R. T. Azuma, "A Survey of Augmented Reality", *Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355-385, August 1997.
- [8] "The Best VR Headsets", *online documentation*, link: <http://www.wareable.com/headgear/the-best-ar-and-vr-headsets>, last accessed 14.04.2017.
- [9] A. Amer, and P. Peralez, "Affordable Altered Perspectives Making Augmented and Virtual Reality Technology Accessible", *Global Humanitarian Technology Conference (GHTC) IEEE*, 2014.
- [10] D. Sharma, "A Review Paper on Virtual Reality Oculus Rift and Augment Reality", *International Journal of Current Research*, vol. 8, no. 09, pp. 37941-37945, September 2016.
- [11] J. Lee, B. Kim, B. Suh, and E. Koh, "Exploring the Front Touch Interface for Virtual Reality Headsets", *2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2585-2591, 2016.
- [12] K. Kim, "Is Virtual Reality (VR) Becoming an Effective Application for the Market Opportunity in Health Care, Manufacturing, and Entertainment Industry?", *European Scientific Journal*, 2016.
- [13] T.J. Dodds, and R.A. Ruddle, "Using Teleporting, Awareness and Multiple Views to Improve Teamwork in Collaborative Virtual Environments", *Visualization and Virtual Reality Research Group School of Computing*, 2008.

- [14] M.S. Hasan, and H. Yu, "Innovative Developments in HCI and Future Trends", *International Journal of Automation and Computing*, vol. 14, no. 1, pp 10-20, December 2016.
- [15] B. Cowan, and B. Kapralos, "An Overview of Serious Game Engines and Frameworks", *Recent Advances in Technologies for Inclusive Well-Being*, vol. 119 of the series Intelligent Systems Reference Library, pp. 15-38, February 2017.
- [16] S. Neelakantam, and T. Pant, *Learning Web-based Virtual Reality: Build and Deploy Web-based Virtual Reality Technology*, Apress, 2017.
- [17] C.-W. Yang, T.-H. Lee, C.-L. Huang, and K.-S. Hsu, "Unity 3D Production and Environmental Perception Vehicle Simulation Platform", *International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, 2016.
- [18] B. Perro, "Sign of a Threat: The Effects of Warning Systems in Survival Horror Games", *COSIGN*, 2004.
- [19] P. Salomoni, C. Prandi, M. Roccetti, L. Casanova, L. Marchetti, and G. Marfia, "Diegetic user Interfaces for Virtual Environments with HMDs: A User Experience Study with Oculus Rift", *Journal on Multimodal User Interfaces*, pp. 1-12, January 2017.